REMARKS

Applicants respectfully request the Examiner's reconsideration of the present application. Applicants believe that consideration of this response could lead to favorable action that would move one or more issues for appeal.

Applicants wish to thank the Examiner for conducting the Examiner's Interview on September 8, 2005.

Claims 1-33 are pending in the present application.

Claims 1-33 are rejected under 35 U.S.C. §102(e) as being unpatentable over U.S. Patent No. 6,817,005 ("Mason").

Claims 1-9, 11-13, 15-19, and 25-33 have been amended.

Support for amended claims 1-9, 11-13, 15-19, and 25-33 is found on pages 4-29 of the specification, Figures 1-17 of the drawings, and claims 1-33 as originally filed. No new matter has been added.

The Examiner states in the Remarks section that

> Applicants argued mainly that Mason does not the broad independent claims. As an example, claim 1 recites generating options... and refining the options... In the remarks such limitations can be found/support in the specification page 4-29. However, the description pages 4-29 cannot be imported into the claimed invention.

(6/22/2005 Office Action, p. 7).

Applicants respectfully submit that Applicants were only citing support for the claims as amended in the Response filed on 4/8/2005 so to point out that no new matter was added with the amendments to the claims. Applicants submit that the claims as presented are believed to be patentable without reading in any additional limitations from the specification.

Claims 1-33 are rejected under 35 U.S.C. §102(e) as being unpatentable over Mason. In particular, with respect to claims 1 and 25, the Examiner states that

> As to claims 1 and 25, Mason et al. teach a method for designing a system on a target device utilizing logic devices (PLD) (see Figs. 1-8 and its description, summary) comprising generating options for utilizing resources on the PLDs in response to user specified constraints (creating unelaborated modules, determining size, location of each unelaborated module to be implemented on PLDs corresponding to determining possible locations to place a component and move the component, user constraints); and refining the options (performing optimizations of placement and routing) for utilizing the resources on the PLDs independent of the user specified constraints (with constraints and without user constraints, see col. 9; the floorplanner tool is used to create position and range (size and shape) constraints for each module, and locate top-level logic as well as module ports and any associated constrained, see at least see summary).

(1/11/2004 Office Action, pp. 2-3).

Applicants respectfully submit that Mason does not render claim 1, as amended, unpatentable under 35 U.S.C. §102(e).

Applicants submit that Mason does not teach or suggest a method for designing a system on a target device utilizing PLDs with an electronic design automation (EDA) tool that includes having the EDA tool determine a first location on the PLD to place a user defined logic region in response to user specified constraints for placement of the user defined logic region, and having the EDA tool determine a second location to place the user defined logic region, wherein the second location is determined independent of the user specified constraints for placement.

In the office action mailed 6/22/2005, the positions presented in the Applicants' response mailed 4/7/2005 were not addressed. Applicants would like to restate that, Mason only discloses a constraint editor tool that is run on the top-level native generic database (NGD) file to generate a top-level user constraint file (UCF) (col. 2, line 66 to col. 3, line 1). In the module build stage, the top-level user constraint file is copied into a module director, creating a module-relative top-level UCF file. A module netlist is created using a standard HDL editor. The module is defined by a module level EDIF file and a module level netlist constraints file (NCF). The ngdbuild tool is run on the module

level EDIF and NCF files as well as the module-relative top-level UCF file and the top-level NGO file to generate a module-level NGO file and a module-relative top-level NGD file (col. 3, line 61 through col. 4, line 3). In the module map stage, a mapper tool is run on the module-relative, top-level NGD file (col. 4, lines10-11).

The Examiner states that Mason teaches "performing optimizations of placement and routing ... ... with constraints and without user constraints" and refers to column 9 for support (1/11/2004 Office Action, pp. 2-3, and 6/22/2005 Office Action, p. 2).

Applicants submit that Column 9 does not support the Examiner's position. Mason states in column 9, lines 55-58 that "The top.ngo file includes the top-level design with any constraints from the top.edn and top.ncf files, but without any constraints from a user constraints file (UCF)." However, Mason then proceeds to describe that the top.ngd file top.ncf file is used together with the top.ngo file.

> In step 108, command ngdbuild-modular initial top.ngo top.ucf triggers the ngdbuild tool to run on the top-level NGO (top.ngo) and the top-level UCF (top.ucf) file to write the constraints of the top-level UCF (top.ucf) file into the top-level NGD (top.ngd) file.

(Mason, col. 10, lines 14-19).

> This top-level UCF (top.ucf) file will be used in the module build stage 200 to correctly position each model. For this reason, the ngdbuild tool does not need to be run to reannotate these constraints back into the top-level NGD (top.ngd) file. Instead, the top.ngo and top.ucf files will be used directly in subsequent stages.

(Mason, col. 12, lines 27-33).

> In step 203, the ngdbuild tool is run on the top-level NGO (top.ngo) file (still located in the TOP/directory), the module level EDIF (A.edn) and NCF (A.ncf) files, and the module-relative top-level UCF file.

(Mason, col. 13, lines 1-4)

> Specifically, in step 801, command ngdbuimodular assembpim_path pim_path –use_pim mouse_pim mouse_pim mode top.ngo triggers the ngdbuild tool to run on the PIM level NGO (A.ngo, B.ngo, and C.ngo) files for each of the implemented modules as well as the top-level NGO (top.ngo) and UCF (top,ucf) files.

(Mason, col. 17, lines 6 through 12).

Thus, Mason describes using user constraints when building modules.

The Examiner also states in the Remarks that

> Based on interpretation of the claimed language, Examiner submits that Mason teaches claimed limitations as recited. The constraints as defined by Mason are described in col. 5. Each module includes associated constraints (routing, timing, area, mapping, placement process). Each module can be designed independently and optionally grouping these modules to generate an IC design utlilizing PAR tools. Mason also teach constraint editor. Utlizing PAR tools and constraint editor, any IC design can be generated.

(6/22/2005 Office Action, pp. 7-8)

Applicants respectfully submit that col. 5, lines 45-51 in Mason describes the definition of "Constraints" and "Constraints Editor". However, the definitions do not teach or suggest having an EDA tool (either a constraints editor or floorplanner tool) determine a first location on the PLD to place a user defined logic region in response to user specified constraints for placement of the user defined logic region, and having the EDA tool determine a second location to place the user defined logic region, wherein the second location is determined independent of the user specified constraints for placement.

On the contrary, as presented above, Mason discloses only describes positioning a module while using the user constraint file (see Mason, col. 12, lines 27-33).

In contrast, amended claim 1 as amended states

> A method for designing a system on a target device utilizing programmable logic devices (PLD) with an electronic automation design tool (EDA), comprising:
> having the EDA tool determine a first location on the PLD to place a user defined logic region in response to user specified constraints for placement of the user defined logic region; and
> having the EDA tool determine a second location to place the user defined logic region, wherein the second location is determined independent of the user specified constraints for placement.

(Claim 1 as amended) (Emphasis added).

Claims 12, and 25, as amended, include similar limitations. Given that claims 2-11, 13-18, and 26-33, depend directly or indirectly from claims 1, 12, and 25, as amended, it is likewise submitted that claims 2-11, 13-18, and 26-23 are also patentable under 35 U.S.C. §102(e) over Mason.

Applicants submit that Mason does not disclose determining routing strategies for routing signals on PLDs in response to user specified routing constraints, and determining additional routing strategies for routing the signals independent of the user specified routing constraints, wherein the routing constraints pertain to a category of routing resources on the PLDs to use.

In contrast, amended claim 19 states

> A method for designing a system on programmable logic devices (PLDs) with an electronic design automation (EDA) tool, comprising:
> having the EDA tool determine routing strategies for routing signals on the PLDs in response to user specified routing constraints that pertain to categories of routing resources to use; and
> having the EDA tool determine additional routing strategies for routing the signals on the PLDs where the additional routing strategies are independent of the user specified routing constraints.

(Claim 19 as amended) (Emphasis added).

Given that claims 20-24, as amended, depend directly or indirectly from claim 19, as amended, it is likewise submitted that claims 20-24 are also patentable under 35 U.S.C. §102(e) over Mason.

In addition, Applicants submit that Mason also fails to disclose having an EDA tool determine the second location in response to the first location not satisfying design parameters or the first location not satisfying the user specified constraints.

The Examiner states

> As to claims 2-3, 16-17 and 26-27, Mason et al. teach refining the options for utilizing the resources is performed in response to the options not satisfying design parameters (performing optimizing of modules placement and routing, modifications allowing fine-tuning the design to meet timing constraints; at least see col. 12-14).

(1/11/2005 Office Action, p. 3).

Applicants submit that columns 12 through 14 describe a module build stage and a module map stage. Columns 12 through 14 do not, however, describe having an EDA tool determine a first location to place a user defined logic region in response to user specified constraints, and having the EDA tool determine a second location to place the user defined logic region independent of the user specified constraints, wherein the second location is determined in response to the first location not satisfying design parameters or the options not satisfying user specified constraints. In fact, column 13 of Mason discloses that user specified constraints are used in the model build stage. "The copied UCF file top.ucf in the current directly is explicitly specified in the argument to the command." (col. 13, lines 22-24).

In contrast, amended claim 2 states

> The method of Claim 1, wherein having the EDA tool determine the second location is performed in response to the first location not satisfying design parameters.

(Claim 2 as amended).

Claims 23, and 26, as amended, include similar limitations.

Claim 3 as amended states

> The method of Claim 1, wherein having the EDA tool determine the second location is performed in response to the first location not satisfying the user specified constraints.

(Claim 3 as amended).

Claims 17, 22, and 27, as amended, include similar limitations.

Furthermore, Applicants submit that Mason also fails to disclose having the EDA tool determine a second location in response to a threshold number of options being generated.

The Examiner states that

> As to claims 4, 18, 24, and 28, Mason et al. teach multiple guides used in placer and router tool, where the placer and router tool iteratively steps through each one and guides the placement and routing of logic corresponding to the logic in the guide file (col. 22). Since the guide file provides complete information to perform placing and routing including optimization, the guide file must include a threshold number of options generated/number of possible locations/number of routing strategies in order to quickly implementing a final product according to user defined constraints.

(1/11/2005 Office Action, pp. 3-4).

Applicants submit that the "multiple guides used" referred to by the Examiner utilize user specified constraints. Mason discloses that "any constraints which are found in the top-level NGD or UCF files are 'pushed' into these guided pieces of logic in a standard manner." (col. 22, lines 31-33). Thus, the guides teach away from determining a second location independent of the user specified constraints. Applicants further submit that Mason fails to disclose that the "multiple guides used" operate in response to a threshold of options generated.

In contrast, amended claim 4 states

> The method of Claim 1, wherein having the EDA tool determine the second location is performed in response to having a threshold number of options generated.

(Claim 4 as amended)

Claims 18, 24, and 28 include similar limitations.

Moreover, Applicants submit that Mason also fails to disclose having the EDA tool determine a second location in response to a triggering event.

The Examiner states that

> As to claims 5 and 29, Mason et al. also teaches triggering event (col. 9, col. 13, col. 17).

(1/11/2005 Office Action, p. 4).

Applicants submit that although columns 9, 13, and 17 include the term "triggers", the term does not refer to an event that causes an EDA tool to determine a

second location for placing a user defined region that is independent of user specified constraints. For example, column 13 of Mason discloses that user specified constraints are used in the model build stage. "The copied UCF file top.ucf in the current directly is explicitly specified in the argument to the command." (col. 13, lines 22-24). Also, column 17 of Mason discloses commands that triggers the ngdbuild tool to run on the PIM level NGO files for each of the implemented "as well as the top-level NGO (top.ngo) and UCF (top.ucf) files." (col. 17, lines 7-12).

In contrast, amended claim 5 states

> The method of Claim 1, wherein having the EDA tool determine the second location is performed in response to a triggering event.

(Claim 5 as amended)

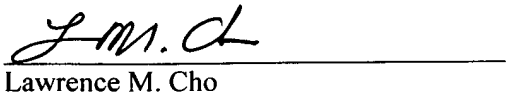
Claims 29 includes similar limitations.

In view of the amendments and arguments set forth herein, it is respectfully submitted that the applicable rejections and have been overcome. Accordingly, it is respectfully submitted that claims 1-33, as amended, should be found to be in condition for allowance.

If any additional fee is required, please charge Deposit Account No. 50-1624.

Respectfully submitted,

Dated: September 15, 2005



Lawrence M. Cho
Attorney for Applicant
Registration No. 39,942

P.O. Box 2144
Champaign, IL 61825
(217) 377-2500